



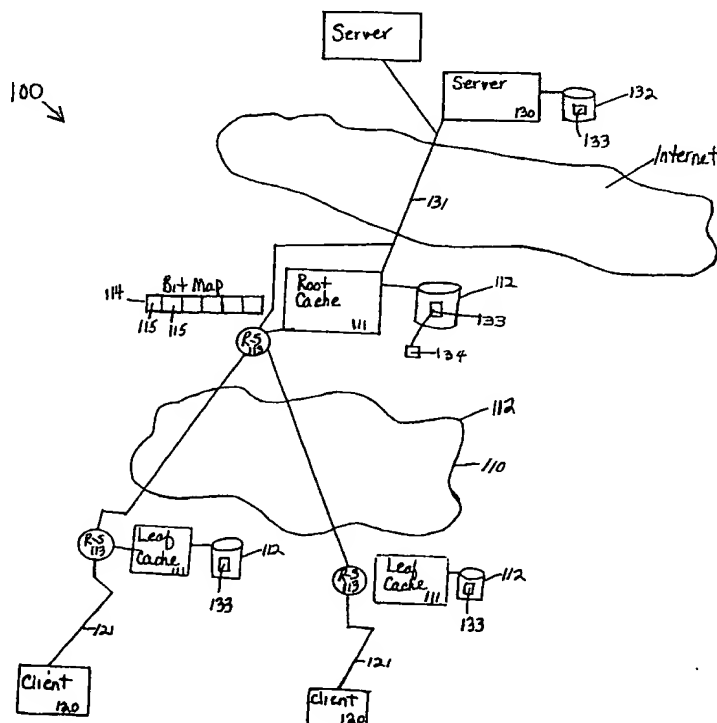
INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification ⁷ : G06F 17/30		A2	(11) International Publication Number: WO 00/07124
			(43) International Publication Date: 10 February 2000 (10.02.00)
(21) International Application Number: PCT/US99/17149 (22) International Filing Date: 28 July 1999 (28.07.99) (30) Priority Data: 09/127,249 31 July 1998 (31.07.98) US (71) Applicant: CACHEFLOW, INC. [US/US]; 650 Almanor Avenue, Sunnyvale, CA 94086 (US). (72) Inventor: MALCOLM, Michael; 521 Shelby Lane, Los Altos, CA 94024 (US). (74) Agent: SWERNOFSKY, Steven, A.; Swernofsky Law Group, P.O. Box 390013, Mountain View, CA 94039-0013 (US).		(81) Designated States: AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZA, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG). Published <i>Without international search report and to be republished upon receipt of that report.</i>	

(54) Title: MULTIPLE CACHE COMMUNICATION

(57) Abstract

The invention provides a method and system for operating multiple communicating caches. Between caches, unnecessary transmission of repeated information is substantially reduced. Each cache maintains information to improve the collective operation of the system of multiple communicating caches. This can include information about the likely contents of each other cache, or about the behavior of client devices or server devices coupled to other caches in the system. Pairs of communicating caches substantially compress transmitted information. This includes both reliable compression, in which the receiving cache can reliably identify the compressed information in response to the message, and unreliable compression, in which the receiving cache will sometimes be unable to identify the compressed information. A first cache refrains from unnecessarily transmitting the same information to a second cache when each already has a copy. This includes both maintaining a record at a first cache of information likely to be stored at a second cache, and transmitting a relatively short identifier for that information in place of the information itself. A set of caches are disposed in a directed graph structure, with a set of root caches disposed for coupling to server devices and a set of leaf caches disposed for coupling to client devices. Both root caches and leaf caches maintain non-cacheable objects beyond their initial use, along with digests of the non-cacheable objects. When a server device returns identical information to a root cache, root caches can transmit only associated digests to leaf caches, avoiding re-transmitting the entire non-cacheable object.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece	ML	Mali	TR	Turkey
BG	Bulgaria	HU	Hungary	MN	Mongolia	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MR	Mauritania	UA	Ukraine
BR	Brazil	IL	Israel	MW	Malawi	UG	Uganda
BY	Belarus	IS	Iceland	MX	Mexico	US	United States of America
CA	Canada	IT	Italy	NE	Niger	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NL	Netherlands	VN	Viet Nam
CG	Congo	KE	Kenya	NO	Norway	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NZ	New Zealand	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	PL	Poland		
CM	Cameroon	KR	Republic of Korea	PT	Portugal		
CN	China	KZ	Kazakstan	RO	Romania		
CU	Cuba	LC	Saint Lucia	RU	Russian Federation		
CZ	Czech Republic	LI	Liechtenstein	SD	Sudan		
DE	Germany	LK	Sri Lanka	SE	Sweden		
DK	Denmark	LR	Liberia	SG	Singapore		
EE	Estonia						

Title of the Invention

MULTIPLE CACHE COMMUNICATION

5

Background of the Invention

1. Field of the Invention

This invention relates to caches.

10

2. Related Art

In a computer system in which client devices request information from one or more server devices, it is sometimes desirable to provide a cache; that is, a device that maintains copies of requested information so multiple requests for the same information can be satisfied at the cache. When requests for information are satisfied at the cache, the server devices need not receive the requests, process them, and retransmit the same information over a communication channel that links the client devices and the server devices. For example, the server devices can be web servers, the client devices can be web clients, the communication channel can be an IP network such as the Internet, and the requested information can be web objects.

Some information requested from the server devices is considered not cacheable, for one or more of several reasons. As examples, the server can refuse to allow the information to be cached, or the information can be a result of a dynamic process that can provide differing results for the same request (so caching would obviate the operation of that dynamic process). An example of dynamically processed information could include advertisements, database searches, or output from CGI scripts.

However, it often occurs that non-cacheable information is requested a second time without having changed, so the second request to the server results in identical information being returned. In a system with multiple communicating caches, transmit-

ting the same information from a first cache to a second cache (when each already has a copy) is an inefficient use of communication resources.

Accordingly, it would be desirable to provide a method and system for operating a set of multiple communicating caches, in which transmission of repeated information is substantially reduced or eliminated. A first aspect of the invention is to maintain information at each cache to improve the collective operation of multiple communicating caches. A second aspect of the invention is to substantially reduce the amount of information transmitted between multiple communicating caches. A third aspect of the invention is to refrain from unnecessarily transmitting the same data from a first cache to a second cache when the latter already has a copy.

Summary of the Invention

The invention provides a method and system for operating a set of multiple communicating caches. Between caches, unnecessary transmission of repeated information is substantially reduced.

In a first aspect of the invention, each cache maintains information to improve the collective operation of the system of multiple communicating caches. This can include information about the likely contents of each other cache, or about the behavior of client devices or server devices coupled to other caches in the system.

In a second aspect of the invention, pairs of communicating caches substantially compress transmitted information. This includes both compression in which the receiving cache can reliably identify the compressed information in response to the message, and compression in which the receiving cache will sometimes be unable to identify the compressed information.

In a third aspect of the invention, a first cache refrains from unnecessarily transmitting the same information to a second cache when each already has a copy. This includes both maintaining a record at a first cache of information likely to be stored at a

second cache, and transmitting a relatively short identifier for that information in place of the information itself.

5 In a preferred embodiment, a set of caches are disposed in a directed graph structure, with a set of root caches disposed for coupling to server devices and a set of leaf caches disposed for coupling to client devices. Both root caches and leaf caches store non-cacheable objects beyond their initial use, along with relatively short identifiers for the non-cacheable objects. When a server device returns identical information to a root cache in response to a request for a non-cacheable object, that root cache transmits
10 only the identifier of the non-cacheable object to the requesting leaf cache, avoiding re-transmitting the entire object if the leaf cache still has the object.

Brief Description of the Drawings

15 Figure 1 shows a block diagram of a system having multiple caches.

Figure 2 shows a process flow diagram for a method of using a system having multiple caches.

Detailed Description of the Preferred Embodiment

20

In the following description, a preferred embodiment of the invention is described with regard to preferred process steps and data structures. Those skilled in the art would recognize after perusal of this application that embodiments of the invention
25 can be implemented using one or more general purpose processors or special purpose processors or other circuits adapted to particular process steps and data structures described herein, and that implementation of the process steps and data structures described herein would not require undue experimentation or further invention.

30 Inventions disclosed herein can be used in conjunction with inventions disclosed in one or more of the following patent applications:

- o Provisional U.S. Application 60/048,986, filed June 9, 1997, in the name of inventors Michael Malcolm and Robert Zarnke, titled "Network Object Cache Engine", assigned to CacheFlow, Inc., attorney docket number CASH-001 (P).
 - 5 o U.S. Application Serial No. 08/959,058, filed October 28, 1997, in the name of inventors Michael Malcolm and Ian Telford, titled "Adaptive Active Cache Refresh", assigned to CacheFlow, Inc., attorney docket number CASH-003.
 - 10 o U.S. Application Serial No. 08/959,313, filed October 28, 1997, in the name of inventors Doug Crow, Bert Bonkowski, Harold Czegledi, and Tim Jenks, titled "Shared Cache Parsing and Pre-fetch", assigned to CacheFlow, Inc., attorney docket number CASH-004.
 - 15 o U.S. Application Serial No. 09/093,533, filed June 8, 1998, in the name of inventors Michael Malcolm and Robert Zarnke, titled "Network Object Cache Engine", assigned to CacheFlow, Inc., attorney docket number CASH-001.
- and
- 20 o PCT International Application PCT/US 98/11834, filed June 9, 1997, in the name of assignee CacheFlow, Inc., and inventors Michael Malcolm and Robert Zarnke, titled "Network Object Cache Engine", attorney docket number CASH-001 PCT.

These applications are referred to herein as the "Cache Disclosures," and are hereby incorporated by reference as if fully set forth herein.

/ / /

System Elements

Figure 1 shows a block diagram of a system having multiple caches.

A system 100 includes a cache system 110, at least one client device 120, and at least one server device 130.

Client Device

Each client device 120 is coupled to the cache system 110 using a client communication path 121. The client communication path 121 can include a dial-up connection, a LAN (local area network), a WAN (wide area network), an ATM network, an IP network (such as an internet, intranet, or extranet), or some combination thereof. In a preferred embodiment, the client communication path 121 includes a dial-up connection, such as for coupling a subscriber to an ISP (internet service provider), or a LAN, such as for coupling a workstation to an internet connection.

As used herein, the terms "client" and "server" refer to relationships between the client or server and the cache 110, not necessarily to particular physical devices.

As used herein, the term "client device" includes any device taking on the role of a client in a client-server environment. There is no particular requirement that the client devices 110 must be individual devices; they can each be a single device, a set of cooperating devices, a portion of a device, or some combination thereof.

Server Device

Each server device 130 is also coupled to the cache system 110 using a server communication path 131. The server communication path 131 can include a dial-up connection, a LAN (local area network), a WAN (wide area network), an ATM network, an IP network (such as an internet, intranet, or extranet), or some combination thereof. In a preferred embodiment, the server communication path 131 includes an internet backbone and an internet connection between the cache system 110 and the internet backbone.

As used herein, the term "server device" includes any device taking on the role of a server in a client-server environment. There is no particular requirement that

the server devices 110 must be individual devices; they can each be a single device, a set of cooperating devices, a portion of a device, or some combination thereof.

The server device 130 includes memory or storage 132 for recording one or more web objects 133. The web objects 133 can include any type of data suitable for transmitting to the client device 110, such as the following:

- o text, color, formatting and directions for display;
 - 10 o pictures, data in graphical formats (such as GIF or JPEG), other multimedia data;
 - o animation, audio (such as streaming audio), movies, and video (such as streaming video), and other data in audio or visual formats (such as MPEG);
 - 15 o program fragments, including applets, Java, JavaScript, and ActiveX; and
 - o other web documents (such as when using frames);
- and
- 20 o other data types (such as indicated by future extensions to HTML, DHTML, SGML, XML, or similar languages).

Cache System

25

The cache system 110 includes a set of caches 111. Each cache 111 is designated a "leaf cache" if it is coupled to one or more client communication paths 121, and is designated a "root cache" if it is coupled to one or more server communication paths 131. The cache system 110 includes an inter-cache communication path 112 for

30 communication between and among caches 111.

The inter-cache communication path 112 can include a plurality of direct connections, a LAN (local area network), a WAN (wide area network), an IP network (such as an internet), or some combination thereof. In a preferred embodiment, the inter-cache communication path 112 includes a plurality of direct connections between pairs of
5 individual caches 111.

In a preferred embodiment, the caches 111 in the cache system 110 are disposed in a graph structure. One or more leaf caches 111 are coupled to client communication paths 121, and one or more root caches 111 coupled to one or more server communication paths 131. Where appropriate, a set of intermediate caches 111 are coupled
10 to the leaf caches 111 and to the root caches 111.

In a preferred embodiment, the graph structure is a tree structure, with a single root cache 111 and a plurality of leaf caches 111. For example, in a cache system
15 110 disposed for use with an ISP (internet service provider), there is one root cache 111 coupled to an internet backbone, and there is one leaf cache 111 for each POP (point of presence). In this example, the inter-cache communication path 112 includes direct connections (such as T1 or T3 connections) between the root cache 111 and each leaf cache
20 111.

Cache Devices

Each cache 111 includes a processor, program and data memory, and memory or storage 112 for recording one or more web objects 133. Each cache 111 re-
25 tains the web objects 133 for repeated serving to client devices 120 in response to web requests.

In a preferred embodiment, each cache 111 includes a router-switch 113, for receiving messages and distinguishing types of messages that should be processed by
30 the cache 111 from those that should not. For example, the router-switch 113 can divert all requests using FTP (file transfer protocol) or HTTP (hypertext transfer protocol) to the cache 111 for processing, while passing through other types of requests unchanged.

In a preferred embodiment, each cache 111 includes a cache device such as described in the Cache Disclosures, hereby incorporated by reference as if fully set forth therein, and is disposed for operating as described therein.

5 Multiple Cache Communication

Each leaf cache 111 receives requests from client devices 120 for web objects 133. The web objects 133 might be cacheable or non-cacheable.

10 If a client device 120 requests a cacheable web object 133, the leaf cache 111 might already have the requested web object 133 in its memory or storage 112. If so, the leaf cache 111 serves the requested web object 133 to the client device 120 without having to request the web object 133 from the root cache 111 or from the server device 130. If the leaf cache 111 does not already have the requested web object 133, the
15 leaf cache 111 requests it from the root cache 111.

The root cache 111 performs a similar caching function, returning the requested cacheable web object 133 directly to the leaf cache 111 if it is already present in its own memory or storage 112, without having to request that web object 133 from the
20 server device 130. If the root cache 111 does not already have the requested web object 133 in its memory or storage 112, the root cache 111 requests it from the server device 120.

If the leaf cache 111 and the root cache 111 do not already have a copy of
25 the web object 133 in their respective memory or storage 112, the root cache 111 requests the web object 133 from the server device 120. Similarly, if the web object 133 is considered not cacheable, the root cache 111 requests the web object 133 from the server device 120 whether or not it has already that web object 133 in their respective memory or storage 112. The server device 120 receives the request and returns the requested web
30 object 133 to the root cache 111.

Objects Already in Storage

The root cache 111 receives the requested web object 133 from the server device 110, records it in its memory or storage 112, and determines an object signature 134 for the web object 133. In a preferred embodiment, the root cache 111 computes the object signature 134 itself. In alternative embodiments, the server device 120 may compute and record the object signature 134 and transmit it to the root cache 111 with the web object 133.

In a preferred embodiment, the object signature 134 includes an MD5 digest of the web object 133. In alternative embodiments, the object signature 134 may comprise a CRC, MD4, SHA, or other known function of the web object 133.

There is no particular need for any device to be able to recover the web object 133 a priori from the object signature 134. It is sufficient that the root cache 111 or the leaf cache 111 can determine, in response to the object signature 134, if the web object 133 is present in its memory or storage 112, and if so, which web object 133 corresponds to that object signature 134.

If the web object 133 is cacheable but was requested from the server device 110, the request from the server device 120 was due to a cache miss. However, it can still occur that the leaf cache 111 (or some intermediate cache 111) already has the web objects 133 in its memory or storage 112, such as recorded in association with a different URL (uniform resource locator) or other identifier. In a preferred embodiment, each cache 111 records web objects 133 in association with the URL used to request those web objects 133.

For a first example, multiple server devices 120 can record mirror copies of identical web objects 133. For a second example, non-identical web objects 133 can include identical embedded web objects 133 (such as common graphics, animation, or program fragments).

If the web object 133 is considered non-cacheable, it was requested from the server device 120 because non-cacheable web objects 133 are not meant to be served from the cache 111. However, it can still occur that the leaf cache 111 (or some intermediate cache 111) already has the web objects 133 in its memory or storage 112, because
5 the non-cacheable web object 133 had been requested earlier.

For a first example, if the web object 133 is responsive to a CGI script or database search, it can be identical to the results of an earlier response to that CGI script or database search. For a second example, if the web object 133 is determined dynamically by the server device 130 (such as randomly selected advertisements), it can be
10 identical to an earlier advertisement transmitted by the server device 130.

The root cache 111 transmits the object signature 134 to the leaf cache 111. The leaf cache 111 determines, in response to the object signature 134, whether it already
15 has the associated web object 133 in its memory or storage 112 and if so, which one is the associated web object 133. If so, the leaf cache 111 serves the associated web object 133 to the client device 120 from its memory or storage 112 without the root cache 111 having to actually transmit the entire web object 133. If not, the root cache 111 transmits the actual web object 133 to the leaf cache 111, which can then serve it to the client de-
20 vice 120.

In a preferred embodiment, the root cache 111 includes a bitmap 114 in its memory or storage 112 for each non-cacheable web object 133, including one bit 115 for each leaf cache 111. Each bit 115 of the bitmap 114 indicates whether its associated leaf
25 cache 111 has a copy of the web object 133.

The root cache 111 directly transmits the actual web object 133 to the leaf cache 111 if the associated bit 115 of the bitmap 114 indicates that the leaf cache 111 does not have the web object 133. If the bit 115 indicates that the leaf cache 111 does
30 have the web object 133, the root cache 111 attempts to transmit only the object signature 134. However, even if the bit 115 indicates that the leaf cache 111 does have the web object 133, it may occur that the leaf cache 111, being a cache, has discarded the web

object 133 in the interim. In this case, the leaf cache 111 so indicates and re-requests the web object 133 from the root cache 111.

In a preferred embodiment, when the root cache 111 transmits the object signature 134 to the leaf cache 111, it so indicates using a data type, such as a MIME type, or a new type of object, indicating that the transmission includes only the object signature 134.

Compression for Transmission

When transmitting actual web objects 133 between caches 111 (such as from the root cache 111 to the leaf cache 111), those web objects 133 are substantially compressed for transmission and decompressed after reception. Compression for transmission can be applied both to cacheable and to non-cacheable web objects 133.

Compression for transmission can include various techniques, such as Huffman coding, Liv-Zempel compression, or other known lossless compression. Compression for transmission can also include known lossy compression, such as JPEG, MPEG, or other audio and video codec techniques, when appropriate for the type of web object 133.

Those skilled in the art will recognize, after perusal of this application, that transmission of the object signature 134 in place of the actual web object 133 is a form of substantial compression. This form of compression is unreliable, in the computer science sense that the receiver is not guaranteed to be able to recover the web object 133 from its object signature 134. In fact, using this form of compression the leaf cache 111 can only do so if the web object 133 is already recorded in its memory or storage 112.

Unreliable Dictionary Compression

As used herein, "dictionary compression" means a form of communication in which a sender and a destination each maintain a set of dictionary elements and a set

of associated tag values, each tag value being representative of one of the dictionary elements. There is no particular requirement that the dictionary elements can be recovered from their associated tag values without further information. Rather, dictionary compression refers generally to a system in which the dictionary elements can be associated with arbitrary tag values.

The sender and the destination each associate the same tag value with the same dictionary element. For example, the sender can transmit the dictionary element, along with an arbitrarily selected tag value, to the destination to make the association. Systems in which the sender does this, and the destination maintains a dictionary of such tag values in response thereto, are known in the art.

As used herein, "unreliable" dictionary compression means that the destination might possibly discard the association between the tag value and the dictionary element.

In a preferred embodiment, each dictionary element includes a complete web object 133, and the tag value associated with each particular web object 133 is a known function of that particular web object 133. The known function is preferably an MD5 signature, as noted herein.

In a preferred embodiment, the destination (because it is a cache) can discard any particular web object 133, and thus lose the association between that particular web object 133 and its MD5 signature. That is, the destination (because it has discarded the particular web object 133) can no longer determine if a particular MD5 signature is associated with any known web object 133. Moreover, the destination cannot determine the web object 133 in response to the MD5 signature without further information.

Transmission of the object signature 134 in place of the actual web object 133 is a form of dictionary compression in which the entire actual web object 133 is the dictionary element. If the leaf cache 111 has discarded that dictionary element, it requests the root cache 111 to retransmit the actual web object 133 using a second form of

compression. For example, the second form of compression can include a known lossless compression technique such as Luv-Zempel compression or the form of compression used in the PKZIP product available from PKWare, Inc.

5 Those skilled in the art will recognize, after perusal of this application, that unreliable dictionary compression is applicable in various other applications that can use compression. In a preferred embodiment, unreliable compression is acceptable because the root cache 111 can retransmit the web object 133 using a more reliable (but possibly less strong) compression technique.

10 Other Web Object Information

The root caches 111 and the leaf caches 111 can also exchange other information about the web objects 133.

15 In a preferred embodiment, the cache system 110 collectively maintains information for each web object 133 regarding the following:

20 o A probability the web object 133 in the cache system 110 will be next requested by some client device 120. This information will likely be best available at the leaf caches 111.

and

25 o A probability the web object 133 in the cache system 110 will be stale. This information will likely be best available at the root caches 111.

30 The cache system 110 can collectively determine from this information whether the web object 133 is the next web object 133 recorded by the cache system 110 to be served state. As described in the Cache Disclosures, particularly attorney docket numbers CASH-003 and CASH-004, this information can be used to determine which web objects 133 to actively refresh.

Active refresh can also be applied to frequently-requested non-cacheable web objects 133, and distributed within the cache system 110, even though those web objects 133 are re-requested from the server devices 120 each time. Active refresh is well suited to web objects 133 such as advertisements, news reports, stock quotes, weather reports, and the like.

The cache system 110 can also maintain information about each web object 133 regarding at which cache 111 in the cache system 110 that web object 133 is recorded. With this information, the root cache 111 can request cached web objects 133 from one of the leaf caches 111, in addition to or instead of re-requesting the web objects 133 from server devices 120.

Method of Operation

Figure 2 shows a process flow diagram for a method of using a system having multiple caches.

A method 200 is performed by the system 100, including the cache system 110, the client devices 120, and the server devices 130.

At a flow point 210, one of the client devices 120 is ready to request a web object 133.

At a step 211, one of the client devices 120 sends a message to its associated leaf cache 111 requesting a selected web object 133. The request message preferably uses the FTP or HTTP protocol, and includes a URL for the selected web object 133.

At a step 212, the leaf cache 111 determines if the web object 133 is cacheable or non-cacheable. If the web object 133 is cacheable, the method 200 proceeds with the next step. If the web object 133 is non-cacheable, the method 200 proceeds with the flow point 220.

At a step 213, the leaf cache 111 determines if the web object 133 is present in its memory or storage 112. In a preferred embodiment, the leaf cache 111 makes this determination in response to the URL for the selected web object 133 included in the request from the client device 120. If the web object 133 is present, the method 200 proceeds with the next step. If the web object 133 is not present, the method 200 proceeds with the flow point 220.

At a step 214, the leaf cache 111 serves the web object 133 to the client device 120. The method 200 continues with the flow point 210.

At a flow point 220, the leaf cache 111 is unable to serve the web object 133 from its memory or storage 112, either because there has been a leaf cache miss or because the web object 133 is non-cacheable.

At a step 221, similar to the step 211, the leaf cache 111 sends a message to the root cache 111 requesting the web object 133.

At a step 222, similar to the step 212, the root cache 111 determines if the web object 133 is cacheable or non-cacheable. If the web object 133 is cacheable, the method 200 proceeds with the next step. If the web object 133 is non-cacheable, the method 200 proceeds with the flow point 230.

At a step 223, similar to the step 213, the root cache 111 determines if the web object 133 is present in its memory or storage 112. In a preferred embodiment, the root cache 111 makes this determination in response to the URL for the selected web object 133 included in the request from the client device 120. If the web object 133 is present, the method 200 proceeds with the next step. If the web object 133 is not present, the method 200 proceeds with the flow point 230.

At a step 224, similar to the step 214, the root cache 111 transmits the web object 133 to the leaf cache 111. The method 200 continues with the flow point 210.

At a flow point 230, the root cache 111 is unable to transmit the web object 133 from its memory or storage 112, either because there has been a root cache miss or because the web object 133 is non-cacheable.

5 At a step 231, similar to the step 211, the root cache 111 sends a message to the indicated server device 130 requesting the web object 133. The request message preferably uses the FTP or HTTP protocol, and includes a URL for the selected web object 133.

10 At a step 232, the server device 130 transmits the web object 133 to the root cache 111.

At a step 233, the root cache 111 determines an object signature 134 for the web object 133.

15 At a step 234, the root cache 111 determines if the web object 133 is present in its memory or storage 112. In a preferred embodiment, the root cache 111 makes this determination in response to the object signature 134. If the web object 133 is present, the method 200 proceeds with the next step. If the web object 133 is not present, the
20 method 200 proceeds with the flow point 240.

At a step 235, the root cache 111 determines if the web object 133 is likely present at the requesting leaf cache 111. In a preferred embodiment, the root cache 111 makes this determination in response to the bitmap 114 for the web object 133. If the
25 web object 133 is likely present at the leaf cache 111, the method 200 proceeds with the next step. If the web object 133 is likely not present at the leaf cache 111, the method proceeds with the flow point 240.

At a step 236, the root cache 111 transmits the object signature 134 to the
30 leaf cache 111.

At a step 237, the leaf cache 111 determines if the web object 133 is present in its memory or storage 112, in response to the object signature 134. If the web object 133 is not present, the method 200 proceeds with the next step. If the web object 133 is present, the method 200 proceeds with the flow point 240.

5

At a step 238, the leaf cache 111 transmits a message to the root cache 111 indicating that the web object 133 is not present.

At a step 239, the root cache 111 transmits the actual web object 133 to the leaf cache 111. As noted above, the actual web object 133 is compressed for transmission and decompressed upon reception.

10

At a flow point 240, the leaf cache 111 is ready to serve the web object 133 to the requesting client device 120. The method proceeds with the step 214.

15

Alternative Embodiments

Although preferred embodiments are disclosed herein, many variations are possible which remain within the concept, scope, and spirit of the invention, and these variations would become clear to those skilled in the art after perusal of this application.

20

Claims

1. A method including steps for
compressing, at a first cache, an object stored at said first cache;
5 coupling a compressed version of said object from said first cache to a second cache.
decompressing, at said second cache, an uncompressed version of said object.
- 10 2. A method as in claim 1, wherein said second cache can only recover an uncompressed object from said compressed object in response to a contents of said second memory.
- 15 3. A method as in claim 1, wherein said second cache can recover an uncompressed object from said compressed object without further information.
4. A method as in claim 1, wherein
said steps for compressing include steps for determining a known function
of a first object at said first cache;
20 said compressed version includes a result of said known function;
said steps for decompressing include steps for determining said known function of a second object at said second cache, and comparing said compressed version with a result of said known function of said second object.
- 25 5. A method as in claim 4, wherein said known function is substantially irreversible.
6. A method including steps for
determining, at a first cache, a first object signature responsive to said first
30 object;
coupling said first object signature to a second cache;

comparing, at said second cache, said first object signature with a function of at least one second object;

refraining from transmitting said first object in response to said steps for comparing.

5

7. A method as in claim 6, including steps for coupling a result of said steps for comparing to said first cache; and coupling said first object to said second cache responsive to said comparison result.

10

8. A method as in claim 6, including steps for storing at said first cache information whether said first object is likely to be stored at said second cache;

15 wherein said steps for transmitting said object signature are responsive to said information.

9. A method including steps for determining, at a first cache, first information regarding a likelihood whether an object received from a server is stale;

20 determining, at a second cache, second information regarding a likelihood whether said object will be requested by a client;

coupling said first information and said second information between said first memory and said second memory;

25 discarding said object at first cache disposed in response to said second information; and

discarding said object at said second cache disposed in response to said first information.

30 10. A method including steps for storing a first object at a first cache; determining an object signature at said first cache in response to said object;

transmitting said object signature to a second cache;
comparing said object signature with a function of at least one object stored
at said second cache;
transmitting said first object to said second cache in response to said steps
5 for comparing.

11. A method as in claim 10,
including steps for storing at said first cache information whether said first
object is likely to be stored at said second cache;
10 wherein said steps for transmitting said object signature are responsive to
said information.

12. A method as in claim 10, including steps for transmitting a result of
said steps for comparing to said first cache.

13. A method including steps for
storing, at a first cache, a first object and first information regarding
whether to discard said first object;
storing, at a second cache, a second object and second information regard-
20 ing whether to discard said second object;
coupling said first information to said second cache; and
discarding said second object in response to said first information and in
response to said second information.

14. A method as in claim 13, including steps for
coupling said second information to said first cache; and
discarding said first object in response to said first information and in re-
sponse to said second information.

15. A method as in claim 13, wherein said first information includes a
likelihood of said object being stored at said second cache.

16. A method including steps for
storing, at a first cache, an object from a server and first information re-
garding said object at said server;
storing, at a second cache, an object for delivery to a client and second in-
5 formation regarding requests for said object by said client;
transmitting said first information from said first cache to said second
cache;
transmitting said second information from said second cache to said first
cache;
10 whereby said first cache and said second cache can collectively operate
more effectively.

17. A system including
a first cache disposed for being coupled to a server, and having memory for
15 storing an object from said server and for storing first information regarding said object
at said server;
a second cache disposed for being coupled to a client, and having memory
for storing said object for delivery to said client and for storing second information re-
garding requests for said object by said client;
20 a first communication path between said first cache and said second cache,
disposed for transmitting said first information from said first cache to said second cache;
a second communication path between said first cache and said second
cache, disposed for transmitting said second information from said second cache to said
first cache;
25 whereby said first cache and said second cache can collectively operate
more effectively.

18. A system including
a first cache disposed for being coupled to a server, and having memory for
30 storing an object from said server, said object having an associated object signature;
a second cache disposed for being coupled to a client, and having memory
for storing said object for delivery to said client;

a communication path between said first cache and said second cache, whereby said first cache can transmit either said object signature or said object to said second cache;

whereby said first cache can refrain from transmitting said object to said second cache when said object is already stored at said second cache.

19. A system as in claim 18, including a communication path between said second cache and said first cache, whereby said second cache can transmit a result of a comparison between said object signature and said object at said second cache.

10

20. A system as in claim 18, including a communication path coupling said object signature to said second cache;

coupling a result of a comparison between said object signature and a function of said object at said second cache; and

15

coupling said object to said second cache responsive to said comparison result.

21. A system as in claim 18, including

20

a first communication path coupling said object signature to said second cache;

a second communication path coupling to said first cache a result of a comparison between said object signature and a function of said object at said second cache; and

25

a third communication path coupling said object to said second cache responsive to said comparison result.

22. A system as in claim 18, including

a third cache disposed for being coupled to a second client, and having memory for storing said object for delivery to said second client;

30

a communication path between said first cache and said third cache, whereby said first cache can transmit either said object signature or said object to said third cache;

whereby said first cache can refrain from transmitting said object to said third cache when said object is already stored at said third cache.

23. A system as in claim 18, including memory at said first cache including information whether said object is likely to be stored at said second cache.

24. A system including
a first cache including a first memory and an information compressor;
a second cache including a second memory and an information decompressor; and
a communication path disposed for coupling a compressed object from said first cache to said second cache.

25. A system as in claim 24, wherein
said first memory includes a first object;
said information compressor includes a known function;
said communication path is disposed for coupling to said second cache a first result of said known function applied to said first object;
said second memory includes a second object;
said information decompressor includes said known function, and a comparison between said first result and a second result of said known function applied to said second object;

whereby said second cache is disposed for determining that said second object is substantially identical to said first object in response to a result of said comparison.

26. A system as in claim 25, wherein said known function is substantially irreversible.

27. A system as in claim 24, wherein said information compressor is lossy.

28. A system as in claim 24, wherein said second cache can only recover an uncompressed object from said compressed object in response to a contents of said second memory.

5 29. A system as in claim 24, wherein said second cache can recover an uncompressed object from said compressed object without further information.

30. A system including
a first cache including a first object and a first object signature responsive
10 to said first object;
a second cache including a second object, said second cache coupled to said first object signature;
said first cache coupled to a result of a comparison between said first object signature and a second object signature responsive to said second object;
15 whereby said first cache can refrain from transmitting said first object in response to said comparison.

31. A system as in claim 30, including a communication path
coupling said first object signature to said second cache;
20 coupling said comparison result to said first cache; and
coupling said first object to said second cache responsive to said comparison result.

32. A system as in claim 30, including
25 a first communication path coupling said first object signature to said second cache;
a second communication path coupling said comparison result to said first cache; and
a third communication path coupling said first object to said second cache
30 responsive to said comparison result.

33. A system as in claim 30, including

a third cache including a third object, said third cache coupled to said first object signature;

said first cache coupled to a result of a second comparison between said first object signature and a third object signature responsive to said third object;

5 whereby said first cache can refrain from transmitting said first object in response to said second comparison.

34. A system as in claim 30, including information at said first cache whether said transmitting said first object is likely to be desirable.

10

35. A system including

a first cache including a first object and first information regarding whether to discard said first object;

15 a second cache coupled to said first information, said second cache including a second object and second information regarding whether to discard said second object, said second cache disposed for discarding said second object in response to said first information and in response to said second information.

36. A system as in claim 35, wherein

20 said first cache is coupled to said second information and is disposed for discarding said first object in response to said first information and in response to said second information.

37. A system as in claim 35, wherein said first information includes a

25 likelihood of said object being stored at said second cache.

38. A system including

a first cache including first information regarding a likelihood whether an object received from a server is stale;

30 a second cache including second information regarding a likelihood whether said object will be requested by a client;

a communication path capable of coupling said first information and said second information between said first memory and said second memory;

said first cache disposed for discarding said object in response to said second information; and

5 said second cache disposed for discarding said object in response to said first information.

39. A system including

10 a first memory including a first object received from a server and a first object signature responsive to said first object;

 a second memory including a second object ready for delivery to a client;

 a communication path capable of coupling said first object signature to said second memory;

15 a comparator coupled to said second memory and to said communication path, responsive to whether a second object signature responsive to said second object is substantially identical to said first object signature;

 said communication path capable of coupling an output of said comparator to said first memory;

20 whereby said first memory can refrain from delivering said first object to said second memory in response to said output of said comparator.

40. A system including

 a plurality of caches, each including memory;

25 a communication network coupling each said cache to substantially all other said caches;

 at least a first said cache having an object in its memory and disposed to transmit said object to a second said cache only if said second cache does not already have said object in its memory.

30 41. A system as in claim 40, including memory at said first cache including information whether said object is likely to be stored at said second cache.

42. A system as in claim 40, wherein said communication network couples an object signature for said object to said second cache; couples to said first cache a result of a comparison between said object signature and a function of said object at said second cache; and
5 couples said object to said second cache responsive to said comparison result.

43. A system as in claim 40, wherein
said first cache includes a first signature function coupled to memory for
10 said first cache;
said second cache includes a second signature function coupled to memory for said second cache, and a comparator coupled to said first signature function and said second signature function;
wherein said first cache refrains from transmitting said first object in re-
15 sponse to said comparator.

44. A system including
a set of root caches disposed for receiving a data item from a set of servers;
a set of leaf caches disposed for delivering said data item to a set of clients,
20 at least one of said leaf caches being coupled to at least one of said root caches;
wherein said root caches and said leaf caches include memory having said data item stored therein after an initial use of said data item;
wherein said data item has an associated item signature, said item signature being responsive to said data item;
25 wherein if a server returns a second data item substantially identical to said one data item, said one root cache transmits said item signature to said one leaf cache;
whereby said one root cache refrains from transmitting said data item more than once to said one leaf cache.

30 45. A system as in claim 44, wherein said set of root caches and said set of leaf caches are disposed in a tree structure.

46. A system as in claim 44, wherein said set of root caches has exactly one member.

47. A system as in claim 44, including memory in said one root cache indicating for each particular said leaf cache whether said particular leaf cache is likely to have said data item in its memory.

48. A method including steps for associating, at both a source and a destination, a tag value with a dictionary element;

if said dictionary element is present at said destination, transmitting, from said source to said destination, said tag value in place of said dictionary element;

if said dictionary element is not present at said destination, transmitting, from said source to said destination, said dictionary element.

49. A method including steps for providing a set of associations, at both a source and a destination, each association including a tag value and a dictionary element;

possibly discarding one or more of said set of associations at said destination;

transmitting, from said source to said destination, a selected tag value or its associated dictionary element, in response to said steps for possibly discarding.

50. A method as in claim 49, including steps for transmitting, from said destination to said source, an indication responsive to said steps for possibly discarding;

whereby if said dictionary element is still present at said destination, said source does not need to transmit said dictionary element to said destination.

51. A method including steps for transmitting a dictionary element from a source to a destination;

associating, at both said source and said destination, a tag value with said dictionary element;

possibly discarding said dictionary element at said destination;

transmitting, from said source to said destination, said tag value or said
5 dictionary element in response to said steps for possibly discarding.

52. A method as in claim 51, including steps for
transmitting, from said destination to said source, an indication responsive
to said steps for possibly discarding;

10 whereby if said dictionary element is still present at said destination, said
source does not need to transmit said dictionary element to said destination.

53. Unreliable dictionary compression.

15 54. A method as in claim 53, wherein a set of dictionary elements for
said method each includes an entire object deliverable from a server to a client.

55. A method as in claim 53, wherein a set of dictionary elements for
said method includes at least one object larger than one kilobyte.

20

56. A system including
means for associating, at both a source and a destination, a tag value with a
dictionary element;

25 means, if said dictionary element is present at said destination, for trans-
mitting, from said source to said destination, said tag value in place of said dictionary
element;

means, if said dictionary element is not present at said destination, for
transmitting, from said source to said destination, said dictionary element.

30

57. A system including
means for providing a set of associations, at both a source and a destina-
tion, each association including a tag value and a dictionary element;

means for possibly discarding one or more of said set of associations at said destination;

means for transmitting, from said source to said destination, a selected tag value or its associated dictionary element, in response to said steps for possibly discarding.

58. A system as in claim 57, including

means for transmitting, from said destination to said source, an indication responsive to said steps for possibly discarding;

whereby if said dictionary element is still present at said destination, said source does not need to transmit said dictionary element to said destination.

59. A system including

means for transmitting a dictionary element from a source to a destination;

means for associating, at both said source and said destination, a tag value with said dictionary element;

means for possibly discarding said dictionary element at said destination;

means for transmitting, from said source to said destination, said tag value or said dictionary element in response to said steps for possibly discarding.

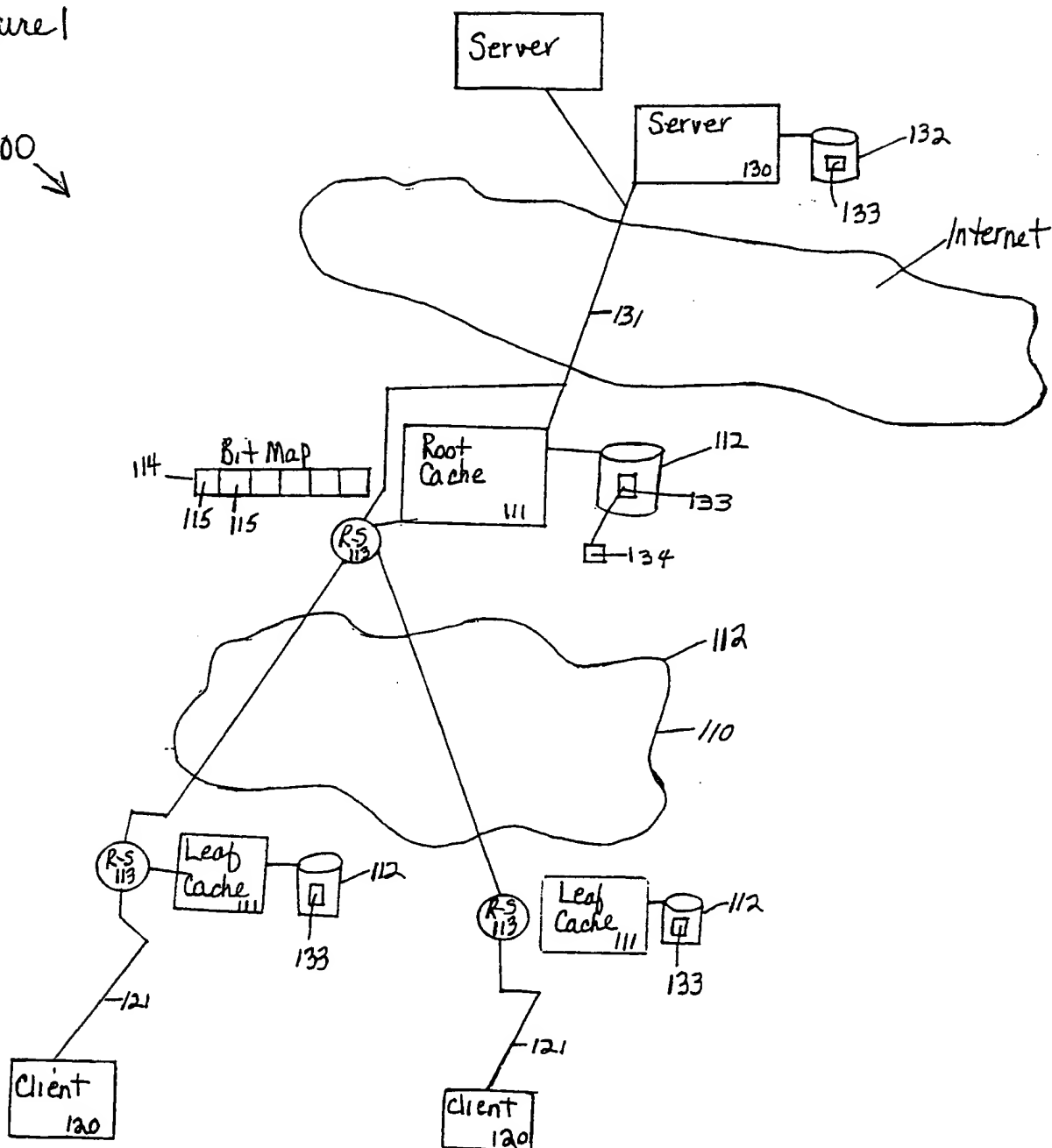
60. A system as in claim 59, including

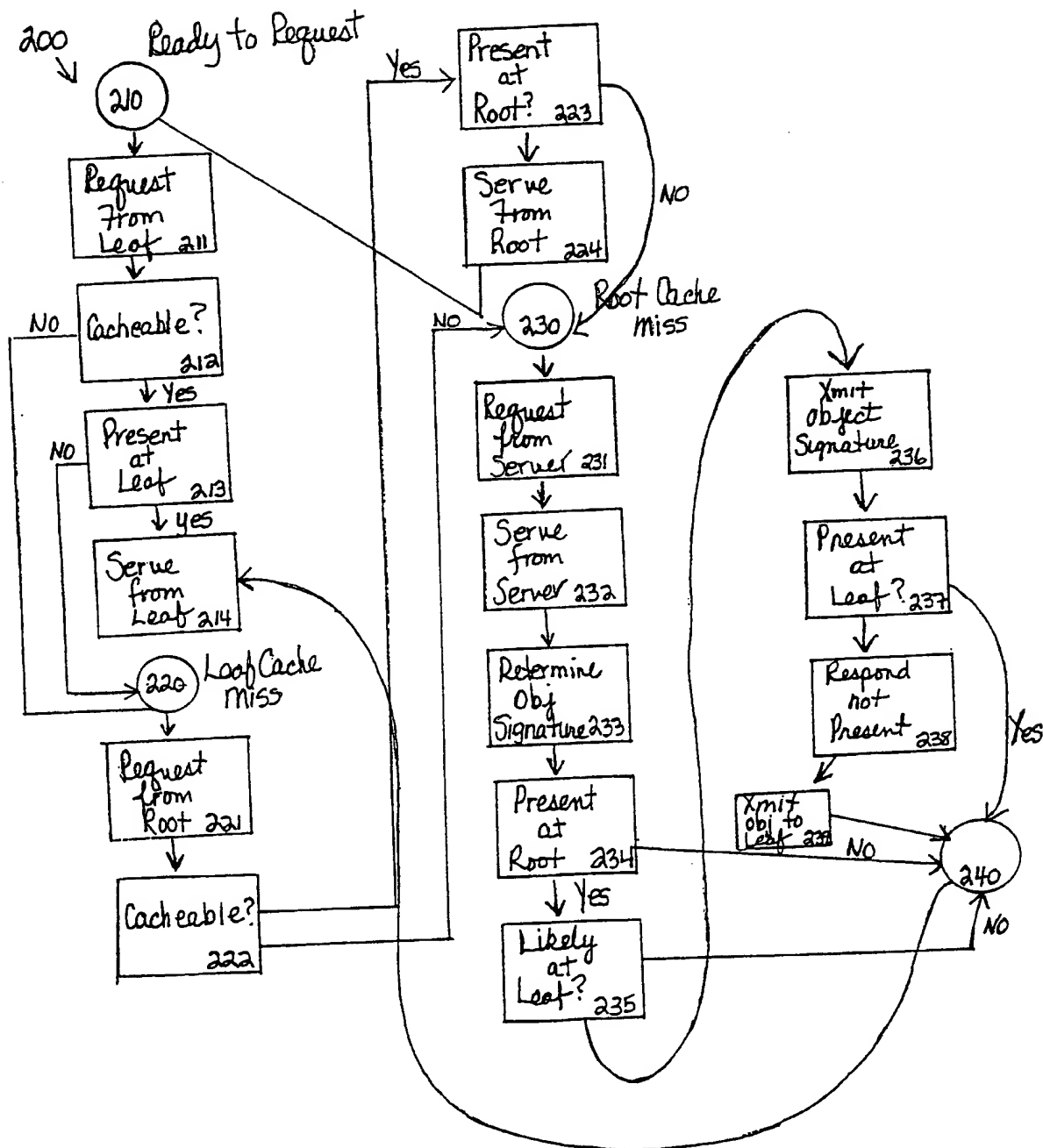
means for transmitting, from said destination to said source, an indication responsive to said steps for possibly discarding;

whereby if said dictionary element is still present at said destination, said source does not need to transmit said dictionary element to said destination.

Cash-007
Figure 1

100 →



Cash-007
Figure 2



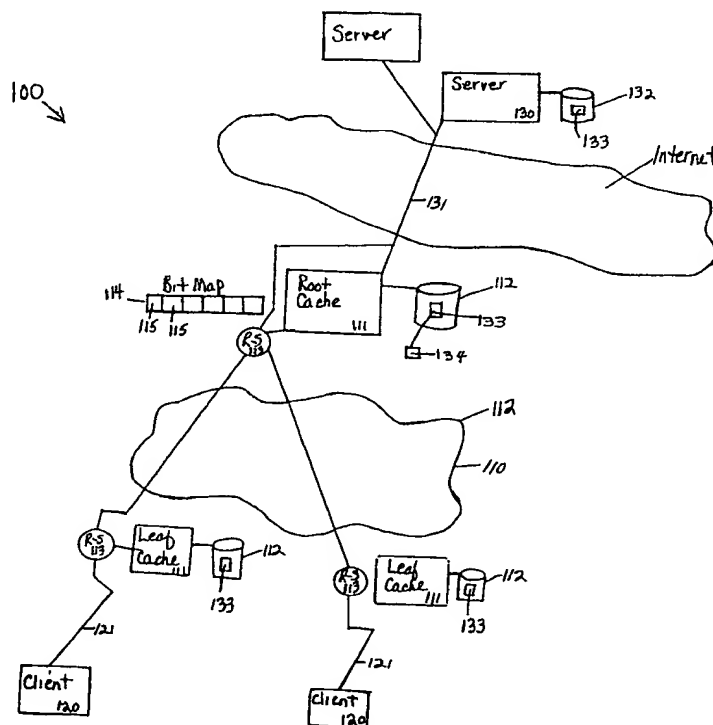
INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

<p>(51) International Patent Classification ⁷ : G06F 17/30, 12/08</p>	<p>A3</p>	<p>(11) International Publication Number: WO 00/07124</p> <p>(43) International Publication Date: 10 February 2000 (10.02.00)</p>
<p>(21) International Application Number: PCT/US99/17149</p> <p>(22) International Filing Date: 28 July 1999 (28.07.99)</p> <p>(30) Priority Data: 09/127,249 31 July 1998 (31.07.98) US</p> <p>(71) Applicant: CACHEFLOW, INC. [US/US]; 650 Almanor Avenue, Sunnyvale, CA 94086 (US).</p> <p>(72) Inventor: MALCOLM, Michael; 521 Shelby Lane, Los Altos, CA 94024 (US).</p> <p>(74) Agent: SWERNOFSKY, Steven, A.; Swernofsky Law Group, P.O. Box 390013, Mountain View, CA 94039-0013 (US).</p>		<p>(81) Designated States: AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZA, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, RU, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).</p> <p>Published <i>With international search report. Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.</i></p> <p>(88) Date of publication of the international search report: 23 March 2000 (23.03.00)</p>

(54) Title: MULTIPLE CACHE COMMUNICATION

(57) Abstract

The invention provides a method and system for operating multiple communicating caches. Between caches, unnecessary transmission of repeated information is substantially reduced. Each cache maintains information to improve the collective operation of the system of multiple communicating caches. This can include information about the likely contents of each other cache, or about the behavior of client devices or server devices coupled to other caches in the system. Pairs of communicating caches substantially compress transmitted information. This includes both reliable compression, in which the receiving cache can reliably identify the compressed information in response to the message, and unreliable compression, in which the receiving cache will sometimes be unable to identify the compressed information. A first cache refrains from unnecessarily transmitting the same information to a second cache when each already has a copy. This includes both maintaining a record at a first cache of information likely to be stored at a second cache, and transmitting a relatively short identifier for that information in place of the information itself. A set of caches are disposed in a directed graph structure, with a set of root caches disposed for coupling to server devices and a set of leaf caches disposed for coupling to client devices. Both root caches and leaf caches maintain non-cacheable objects beyond their initial use, along with digests of the non-cacheable objects. When a server device returns identical information to a root cache, root caches can transmit only associated digests to leaf caches, avoiding re-transmitting the entire non-cacheable object.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece	ML	Mali	TR	Turkey
BG	Bulgaria	HU	Hungary	MN	Mongolia	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MR	Mauritania	UA	Ukraine
BR	Brazil	IL	Israel	MW	Malawi	UG	Uganda
BY	Belarus	IS	Iceland	MX	Mexico	US	United States of America
CA	Canada	IT	Italy	NE	Niger	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NL	Netherlands	VN	Viet Nam
CG	Congo	KE	Kenya	NO	Norway	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NZ	New Zealand	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	PL	Poland		
CM	Cameroon		Republic of Korea	PT	Portugal		
CN	China	KR	Republic of Korea	RO	Romania		
CU	Cuba	KZ	Kazakstan	RU	Russian Federation		
CZ	Czech Republic	LC	Saint Lucia	SD	Sudan		
DE	Germany	LI	Liechtenstein	SE	Sweden		
DK	Denmark	LK	Sri Lanka	SG	Singapore		
EE	Estonia	LR	Liberia				

INTERNATIONAL SEARCH REPORT

International Application No

PCT/US 99/17149

A. CLASSIFICATION OF SUBJECT MATTER

IPC 7 G06F17/30 G06F12/08

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC 7 G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	<p>WO 97 30539 A (IBM ;BITTINGER REED RICHARD (US); FRAENKEL MICHAEL LEVI (US); HOUS) 21 August 1997 (1997-08-21)</p> <p>page 14, line 15 -page 28, line 24; figures 2-10</p> <p style="text-align: center;">--- -/--</p>	<p>1-8, 10-12, 18-34, 39-52, 56-60</p>

☒ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

* Special categories of cited documents :

- "A" document defining the general state of the art which is not considered to be of particular relevance
- "E" earlier document but published on or after the international filing date
- "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
- "O" document referring to an oral disclosure, use, exhibition or other means
- "P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search

28 January 2000

Date of mailing of the international search report

10/02/2000

Name and mailing address of the ISA

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,
Fax: (+31-70) 340-3016

Authorized officer

Nielsen, O

INTERNATIONAL SEARCH REPORT

International Application No

PCT/US 99/17149

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	<p>BANGA G ET AL: "Optimistic deltas for WWW latency reduction"</p> <p>PROCEEDINGS OF THE USENIX 1997 ANNUAL TECHNICAL CONFERENCE, PROCEEDINGS OF USENIX 1997 ANNUAL TECHNICAL CONFERENCE, ANAHEIM, CA, USA, 6-10 JAN. 1997, 6 - 10 January 1997, pages 289-303, XP002129102</p> <p>1997, Berkeley, CA, USA, USENIX Assoc, USA</p> <p>ISBN: 1-880446-84-7</p> <p>page 295, left-hand column, line 5 -page 296, right-hand column</p> <p>page 292, right-hand column, line 28 - line 41</p> <p style="text-align: center;">---</p>	<p>1-8, 10-12, 18-34, 39-52, 56-60</p>
X	<p>YU P S ET AL: "Performance study of a collaborative method for hierarchical caching in proxy servers"</p> <p>COMPUTER NETWORKS AND ISDN SYSTEMS,NL,NORTH HOLLAND PUBLISHING. AMSTERDAM,</p> <p>vol. 30, no. 1-7,</p> <p>1 April 1998 (1998-04-01), pages 215-224, XP004121425</p> <p>ISSN: 0169-7552</p> <p>page 216, right-hand column, line 23 -page 218, left-hand column, line 17</p> <p style="text-align: center;">---</p>	<p>9,13-17, 35-38,49</p>
X	<p>EP 0 836 145 A (AT & T CORP)</p> <p>15 April 1998 (1998-04-15)</p> <p>the whole document</p> <p style="text-align: center;">---</p>	<p>1-52, 56-60</p>
P,X	<p>US 5 864 837 A (MAIMONE WILLIAM D)</p> <p>26 January 1999 (1999-01-26)</p> <p style="text-align: center;">---</p> <p>column 1, line 58 -column 2, line 25</p> <p style="text-align: center;">-----</p>	<p>1-7,10, 18-22, 24-26, 28-34, 40-43</p>

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No

PCT/US 99/17149

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 9730539 A	21-08-1997	US 5859971 A	12-01-1999
		CA 2218187 A	21-08-1997
		CN 1184575 A	10-06-1998
		CZ 9703540 A	18-03-1998
		EP 0823171 A	11-02-1998
		HU 9801874 A	30-11-1998
		JP 11500250 T	06-01-1999
		PL 322830 A	16-02-1998
EP 0836145 A	15-04-1998	US 5931904 A	03-08-1999
		CA 2217370 A	11-04-1998
		JP 10214239 A	11-08-1998
US 5864837 A	26-01-1999	NONE	

THIS PAGE BLANK (USPTO)